Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method in a data processing system for processing instructions, the method comprising:

responsive to receiving an instruction at a processor in the data processing system, determining whether an indicator is associated with the instruction, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;

enabling counting, by the processor, of each event associated with execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the events associated with execution of the instruction in a hardware counter;

determining if the count of the events associated with the execution of the instruction stored in the hardware counter meets or exceeds a threshold; and

identifying a portion of code associated with the instruction as being a hot spot if the count of the events associated with the execution of the instruction in the hardware counter meets or exceeds the threshold, wherein identifying a portion of code associated with the instruction as being a hot spot comprises:

generating, in the processor, an interrupt; and

sending the interrupt to an interrupt handler of a performance monitoring application.

2. (Original) The method of claim 1, wherein the instruction is received in an instruction cache in the processor.

3. (Original) The method of claim 1, wherein the indicator is stored in a performance instrumentation shadow cache and wherein the processor checks the performance instrumentation shadow cache to determine whether the indicator is associated with the instructions.

4. (Original) The method of claim 1, wherein the instruction is received in a bundle by an instruction cache in the processor and wherein the indicator comprises at least one spare bit in a field in the bundle.

5. (Original) The method of claim 1, wherein the indicator is a separate instruction.

6. (Original) The method of claim 1, wherein an event in the events includes at least one of an entry into a module, an exit from a module, an entry into a subroutine, an exit from a subroutine, an entry into a function, an exit from a function, a start of input/output, a completion of input/output, and the execution of the instruction.

7. (Original) The method of claim 1, wherein the determining step comprises:

determining, by an instruction cache, whether the indicator is present in a field within the instruction.

8. (Currently amended) The method of claim 1, wherein the enabling step comprises:

sending a signal to [[a]] the performance monitor unit, wherein the performance monitor unit counts each event associated with execution of the instruction using the hardware counter.

9. (Canceled)

10. (Currently amended) The method of claim [[9]] 1, wherein the performance monitoring application, upon receiving the interrupt, performs an action associated with the identification of a hot spot in the instructions.

11. (Original) The method of claim 10, wherein the action includes:

storing the portion of code corresponding to the hot spot in a shadow data structure; and

generating a mapping from old addresses associated with the portion of code to new addresses of the portion of code in the shadow data structure.

12. (Original) The method of claim 10, wherein the action includes at least one of generating a performance monitoring application log entry and notifying a log daemon process.

13. (Currently amended) A computer program product in a recordable-type computer readable medium for processing instructions comprising:

first instructions for determining whether an indicator is associated with an instruction in response to receiving the instruction at a processor in the data processing system, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;

second instructions for enabling counting, by the processor, of each event associated with execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the events associated with execution of the instruction in a hardware counter;

third instructions for determining if the count of the events associated with the execution of the instruction stored in the hardware counter exceeds a threshold; and

fourth instructions for identifying a portion of code associated with the instruction as being a hot spot if the count of the events associated with the execution of the instruction in the hardware counter exceeds the threshold, wherein the fourth instructions for identifying a portion of code associated with the instruction as being a hot spot include:

instructions for generating, in the processor, an interrupt; and

instructions for sending the interrupt to an interrupt handler of a performance monitoring application.

14. (Original) The computer program product of claim 13, wherein the instruction is received in an instruction cache in the processor.

15. (Original) The computer program product of claim 13, wherein the indicator is stored in a performance instrumentation shadow cache and wherein the performance instrumentation shadow cache is checked to determine whether the indicator is associated with the instructions.

16. (Original) The computer program product of claim 13, wherein the instruction is received in a bundle by an instruction cache in the processor and wherein the indicator comprises at least one spare bit in a field in the bundle.

17. (Original) The computer program product of claim 13, wherein the indicator is a separate instruction.

18. (Original) The computer program product of claim 13, wherein an event in the events includes at least one of an entry into a module, an exit from a module, an entry into a subroutine, an exit from a subroutine, an entry into a function, an exit from a function, a start of input/output, a completion of input/output, and the execution of the instruction.

19. (Original) The computer program product of claim 13, wherein the first instructions include instructions for determining, by an instruction cache, whether the indicator is present in a field within the instruction.

20. (Currently amended) The computer program product of claim 13, wherein the second instructions for enabling include instructions for sending a signal to [[a]] the performance monitor unit, wherein the performance monitor unit counts each event associated with execution of the instruction using the hardware counter.

21. (Canceled)

22. (Currently amended) The computer program product of claim [[21]] 13, wherein the performance monitoring application, upon receiving the interrupt, performs an action associated with the identification of a hot spot in the instructions.

23. (Original) The computer program product of claim 22, wherein the action includes:

storing the portion of code corresponding to the hot spot in a shadow data structure; and

generating a mapping from old addresses associated with the portion of code to new addresses of the portion of code in the shadow data structure.

24. (Original) The computer program product of claim 22, wherein the action includes at least one of generating a performance monitoring application log entry and notifying a log daemon process.

25. (Currently amended) An apparatus for processing instructions comprising:

means for determining whether an indicator is associated with an instruction in response to receiving the instruction at a processor in the data processing system, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;

means for enabling counting, by the processor, of each event associated with execution of the instruction if the indicator is associated with the instruction, wherein the processor autonomically increments the count of the events associated with execution of the instruction in a hardware counter;

means for determining if the count of the events associated with the execution of the instruction stored in the hardware counter exceeds a threshold; and

means for identifying a portion of code associated with the instruction as being a hot spot if the count of the events associated with the execution of the instruction in the hardware counter exceeds the threshold.